Chapter 3: Introduction to SQL

Database System Concepts, 7th Ed.

Nested Subqueries

- SQL provides a mechanism for the nesting of subqueries. A **subquery** is a **select-from-where** expression that is nested within another query.
- The nesting can be done in the following SQL query

  select A_1, A_2, ..., A_n
  from r_1, r_2, ..., r_m
  where P

  as follows:

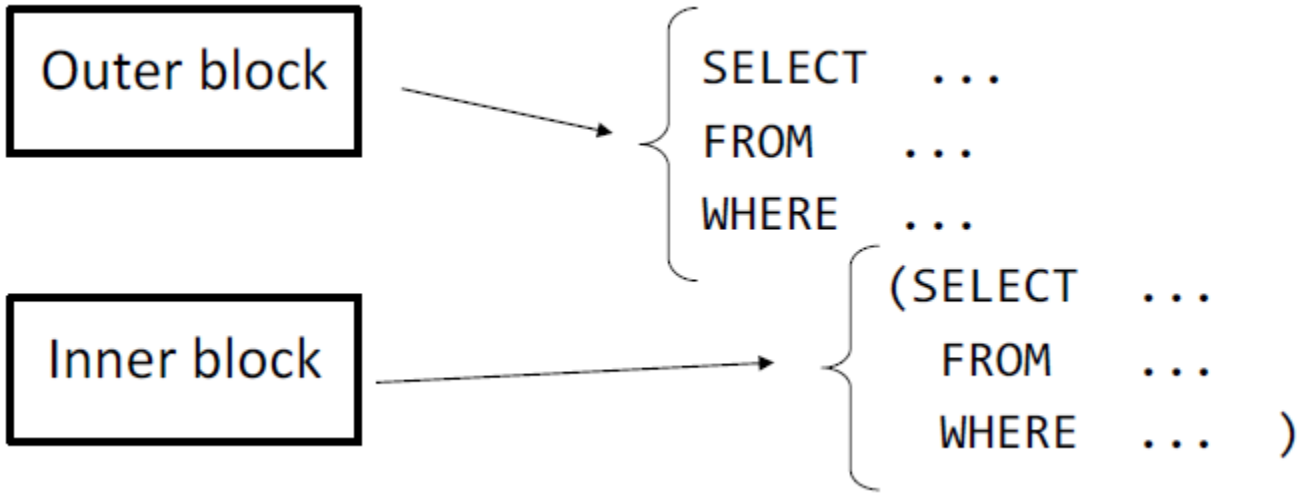
  - **From clause:** r_i can be replaced by any valid subquery
  - **Where clause:** P can be replaced with an expression of the form:

    B <operation> (subquery)

    B is an attribute and <operation> to be defined later.
  - **Select clause:**

    A_i can be replaced be a subquery that generates a single value.

Nested Subqueries



Set Membership

- Find courses offered in Fall 2017 and in Spring 2018

```
select distinct course_id
from section
where semester = 'Fall' and year= 2017 and
        course_id in (select course_id
                      from section
                      where semester = 'Spring' and year= 2018);
```

- Find courses offered in Fall 2017 but not in Spring 2018

```
select distinct course_id
from section
where semester = 'Fall' and year= 2017 and
        course_id  not in (select course_id
                           from section
                           where semester = 'Spring' and year= 2018);
```

Set Membership (Cont.)

- Name all instructors whose name is neither “Mozart” nor Einstein”

```
select distinct name
from instructor
where  name not in ('Mozart', 'Einstein')
```

- Find the total number of (distinct) students who have taken course sections taught by the instructor with *ID* 10101

```
select count (distinct ID)
from takes
where (course_id, sec_id, semester, year) in
                    (select course_id, sec_id, semester, year
                     from teaches
                     where teaches.ID= 10101);
```

- Note: Above query can be written in a much simpler manner. The formulation above is simply to illustrate SQL features

Set Membership (Cont.)

- IN en situaciones es similar a INTERSECT y / o Join
- NOT IN en situaciones es similar a MINUS

Liste los estudiantes que han presentado exámenes, esto es, tienen resultados

```
SELECT s.*
FROM
   students s
WHERE
   s.sid IN
   (
      SELECT sid
      FROM    results
   )
```

SID	FIRST	LAST	EMAIL	GRP	MONITOR
101	Ann	Smith	a@jav.co.c	1	(null)
102	Michael	Jones	(null)	2	103
103	Richard	Turner	b@jav.co.c	2	101

```
SELECT *
FROM
    results    b,
    students   a
WHERE
    a.sid = b.sid
```

SID	CAT	ENO	POINTS	SID_1	FIRST	LAST	EMAIL	GRP	MONITOR
101	H	1	10	101	Ann	Smith	a@jav.co.c	1	(null)
101	H	2	8	101	Ann	Smith	a@jav.co.c	1	(null)
101	M	1	12	101	Ann	Smith	a@jav.co.c	1	(null)
102	H	1	9	102	Michael	Jones	(null)	2	103
102	M	1	10	102	Michael	Jones	(null)	2	103
102	H	2	9	102	Michael	Jones	(null)	2	103
103	H	1	5	103	Richard	Turner	b@jav.co.c	2	101
103	M	1	7	103	Richard	Turner	b@jav.co.c	2	101

Set Membership (Cont.)

- IN en situaciones es similar a INTERSECT y / o Join
- NOT IN en situaciones es similar a MINUS

Liste los estudiantes que NO han presentado exámenes, esto es, tienen resultados

```
SELECT s.*
FROM
  students s
WHERE
  s.sid NOT IN
  (
    SELECT sid
    FROM   results
  )
```

```
SELECT   sid
FROM   students
MINUS
SELECT   sid
FROM   results;
```

SID	FIRST	LAST	EMAIL	GRP	MONITOR
104	Ann	Brown	ca@jav.co.c	3	103

SID
104